IN THE CLAIMS

1. – 53 (Canceled)

54. (Previously Presented)   A machine implemented method performed by a network element having a first interface communicatively coupled to a subscriber over a network provider network and a second interface communicatively coupled to a service provider over a service provider network, the method comprising:

receiving a request from a user via a command line interface (CLI) of the network element for configuring the network element, the request accessing a configuration file stored in a database that is used to route network traffic between the network provider network and the service provider network via the first and second interfaces, the network provider network being different than the service provider network;

in response to the request, recording operations of the request in a transaction log separated from the database without accessing the database until a commit command is received from the user via the CLI of the network; and

performing the operations of the request from the transaction log to access a record of the database associated with the request in response to a commit command from the CLI indicating that the user has committed to the requested configuration.

55. (Previously Presented)   The method of claim 54, wherein the transaction log comprises a persistent memory in which content of the transaction log is maintained after the network element is powered down or rebooted.

56. (Previously Presented)   The method of claim 54, further comprising prior to recording the operations of the request in the transaction log, acquiring a lock for locking the record of the database associated with the request to prevent other users from accessing the record of the database.

57. (Previously Presented)   The method of claim 56, further comprising:

   receiving further modification of configuration from the user prior to the commit
      command; and
   storing the modification in the transaction log without accessing the data base until the
      commit command is received from the user upon which the modification of the
      configuration is committed from the transaction log to the locked record of
      database.

58. (Previously Presented)   The method of claim 56, further comprising:

   receiving an abort command from the user via the CLI prior to receiving the commit
      command; and
   in response to the abort command, removing the operations of the request from the
      transaction log and releasing the acquired lock without accessing the database.

59. (Previously Presented)   The method of claim 58, wherein after performing the removing

and releasing in response to the abort command, the record of the database remains

substantially identical with respect to the record prior to receiving the request.

60. (Previously Presented)   The method of claim 56, further comprising indicating within

the transaction log that the request is in a committing state wile committing the operations of

the request from the transaction log to the locked record of the database.

61. (Previously Presented)   The method of claim 60, further comprising indicating within

the transaction log that the request is in a non-transaction state if operations of committing the

operations of the request from the transaction log to the database have completed.

62. (Previously Presented)   The method of claim 61, further comprising indicating within

the transaction log that the request is in a transaction state while recording the operations of

the request in the transaction log before receiving the commit command from the user.

63. (Previously Presented)   The method of claim 62, further comprising:

     detecting whether operations of committing the operations of the request from the

          transaction log to the database have stopped resulted from errors of the

          network element; and

     in response to the detection, renewing performing the operations of the request from

          the transaction log to the database while the record of the database is locked.

64. (Previously Presented)   The method of claim 63, wherein the detection of whether operations of committing the operations of the request has stopped resulted from errors is performed in response to the network element crashes and recovers from the crash.

65. (Previously Presented)   The method of claim 64, wherein the detection is performed by examining within the transaction log whether the request is in the committing state, and wherein the renewing is performed only if the request is in the committing state.

66. (Previously Presented)   The method of claim 62, further comprising:

   detecting whether operations of recording the operations of the request within the

      transaction log have stopped resulted from errors of the network element; and

   in response to the detection, removing the request from the transaction log without

      committing to the database.

67. (Previously Presented)   The method of claim 66, wherein the detection of whether operations of recording the operations of the request within the transaction log has stopped resulted from errors is performed in response to the network element crashes and recovers from the crash.

68. (Previously Presented)   The method of claim 67, wherein the detection is performed by examining within the transaction log whether the request is in the transaction state, and wherein the removing is performed only if the request is in the transaction state.

69. (Previously Presented)   The method of claim 56, further comprising:

determining whether the lock being acquired is unavailable;

notifying the user via the CLI that the lock is unavailable; and

prompting the user whether the user desires to wait or cancel the request.

70. (Previously Presented)   The method of claim 59, further comprising:

removing the request from the transaction log in response to receiving a cancel

command from the user in response to the prompting; and

in response to receiving a wait command from the user, repeating acquiring the lock

until the lock has been acquired upon which if the commit command has been

received, the request is committed from the transaction log to the locked record

of the database.

71. (Previously Presented)   A machine-readable medium having executable code to cause a

machine to perform a method of a network element having a first interface communicatively

coupled to a subscriber over a network provider network and a second interface

communicatively coupled to a service provider over a service provider network, the method

comprising:

receiving a request from a user via a command line interface (CLI) of the network

element for configuring the network element, the request accessing a

configuration file stored in a database that is used to route network traffic

between the network provider network and the service provider network via the

first and second interfaces, the network provider network being different than

the service provider network;

in response to the request, recording operations of the request in a transaction log

separated from the database without accessing the database until a commit

command is received from the user via the CLI of the network; and

performing the operations of the request from the transaction log to access a record of

the database associated with the request in response to a commit command

from the CLI indicating that the user has committed to the requested

configuration.

72. (Previously Presented)   The machine-readable medium of claim 71, wherein the

transaction log comprises a persistent memory in which content of the transaction log is

maintained after the network element is powered down or rebooted.

73. (Previously Presented)   The machine-readable medium of claim 71, wherein the method

further comprises prior to recording the operations of the request in the transaction log,

acquiring a lock for locking the record of the database associated with the request to prevent

other users from accessing the record of the database.

74. (Previously Presented)   The machine-readable medium of claim 73, wherein the method

further comprises:

receiving further modification of configuration from the user prior to the commit

command; and

storing the modification in the transaction log without accessing the data base until the

commit command is received from the user upon which the modification of the

configuration is committed from the transaction log to the locked record of database.

75. (Previously Presented) The machine-readable medium of claim 73, wherein the method further comprises:

receiving an abort command from the user via the CLI prior to receiving the commit command; and

in response to the abort command, removing the operations of the request from the transaction log and releasing the acquired lock without accessing the database.

76. (Previously Presented) The machine-readable medium of claim 75, wherein after performing the removing and releasing in response to the abort command, the record of the database remains substantially identical with respect to the record prior to receiving the request.

77. (Previously Presented) The machine-readable medium of claim 73, wherein the method further comprises indicating within the transaction log that the request is in a committing state wile committing the operations of the request from the transaction log to the locked record of the database.

78. (Previously Presented) The machine-readable medium of claim 77, wherein the method further comprises indicating within the transaction log that the request is in a non-transaction state if operations of committing the operations of the request from the transaction log to the database have completed.

Application Serial No. 09/873,730   -8-   Atty. Docket No. 4906.P021

79. (Previously Presented)   The machine-readable medium of claim 78, wherein the method further comprises indicating within the transaction log that the request is in a transaction state while recording the operations of the request in the transaction log before receiving the commit command from the user.

80. (Previously Presented)   The machine-readable medium of claim 79, wherein the method further comprises:

    detecting whether operations of committing the operations of the request from the

        transaction log to the database have stopped resulted from errors of the

        network element; and

    in response to the detection, renewing performing the operations of the request from

        the transaction log to the database while the record of the database is locked.

81. (Previously Presented)   The machine-readable medium of claim 80, wherein the detection of whether operations of committing the operations of the request has stopped resulted from errors is performed in response to the network element crashes and recovers from the crash.

82. (Previously Presented)   The machine-readable medium of claim 81, wherein the detection is performed by examining within the transaction log whether the request is in the committing state, and wherein the renewing is performed only if the request is in the committing state.

Application Serial No. 09/873,730   -9-   Atty. Docket No. 4906.P021

83. (Previously Presented)   The machine-readable medium of claim 79, wherein the method further comprises:

    detecting whether operations of recording the operations of the request within the

        transaction log have stopped resulted from errors of the network element; and

    in response to the detection, removing the request from the transaction log without

        committing to the database.

84. (Previously Presented)   The machine-readable medium of claim 83, wherein the detection of whether operations of recording the operations of the request within the transaction log has stopped resulted from errors is performed in response to the network element crashes and recovers from the crash.

85. (Previously Presented)   The machine-readable medium of claim 84, wherein the detection is performed by examining within the transaction log whether the request is in the transaction state, and wherein the removing is performed only if the request is in the transaction state.

86. (Previously Presented)   The machine-readable medium of claim 83, wherein the method further comprises:

    determining whether the lock being acquired is unavailable;

    notifying the user via the CLI that the lock is unavailable; and

    prompting the user whether the user desires to wait or cancel the request.

Application Serial No. 09/873,730   -10-   Atty. Docket No. 4906.P021

87. (Previously Presented)   The machine-readable medium of claim 86, wherein the method further comprises:

removing the request from the transaction log in response to receiving a cancel

command from the user in response to the prompting; and

in response to receiving a wait command from the user, repeating acquiring the lock

until the lock has been acquired upon which if the commit command has been

received, the request is committed from the transaction log to the locked record

of the database.

88. (Canceled)